

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS

1. (Previously Presented) A method of specifying a stateful web service comprising:
first facilitating, by an integrated development environment of a computing device, a user in providing a source code representation of at least a portion of web service logic, the logic including one or more methods;
second facilitating, by the integrated development environment of the computing device, the user in identifying one of said one or more methods to be exposed as part of the stateful web service; and
in response to user input, automatically specifying, by the integrated development environment of the computing device, one or more declarative annotations within the source code representation, the declarative annotations, when recognized by a compiler through analysis of the represented portion of web service logic which includes the declarative annotations, causing the compiler to generate one or more persistent components to maintain conversational state related to the identified method;
wherein the one or more declarative annotations indicate to the compiler whether the identified method is at least one of a start method, a continue method, or a finish method, wherein the start method applies to start of a stateful conversation between a client and the web service, the continue method applies to continuation of an ongoing stateful conversation between a client and the web service, and the finish method applies to completion of an ongoing stateful conversation between a client and the web service.
2. (Cancelled)
3. (Previously Presented) The method of claim 1, wherein the one or more declarative annotations are specified within a comment field preceding the identified method.

4.-5. (Cancelled)

6. (Previously Presented) The method of claim 1, wherein when a method declared to be a start method is invoked at run-time, a new instance of a conversation is created, and a unique identifier is associated with that conversational instance to facilitate management of multiple simultaneous conversations.

7. (Previously Presented) The method of claim 1, wherein when a method declared to be a continue method or a finish method is invoked at run-time, a unique identifier provided by the client is obtained and used to access a corresponding instance of a conversation.

8. (Original) The method of claim 7, wherein when a finish method is invoked at run-time, the corresponding instance of the conversation is destroyed after processing by the web service logic.

9. (Original) The method of claim 1, wherein the one or more declarative annotations indicate to the compiler whether the identified method is buffered, wherein if the identified method is buffered the compiler instantiates one or more queues to temporarily store one or more requests for the identified method.

10. (Previously Presented) The method of claim 1, wherein an additional one or more declarative annotations are manually specified by a developer.

11. (Cancelled)

12. (Previously Presented) The method of claim 1, wherein said user input includes graphical manipulation of the identified method by the user via the integrated development environment.

13. (Original) The method of claim 1, wherein the one or more declarative annotations cause the compiler to generate a proxy object designed to facilitate interaction by the web service with one of an external web service or client.

14. (Original) The method of claim 13, wherein the one or more declarative annotations further cause the compiler to route asynchronous responses from the external web service to code specified by a developer of the web service.

15. (Original) The method of claim 13, wherein the one or more declarative annotations further cause the compiler to generate a unique identifier to identify a specific conversational instance of the external service.

16. (Currently Amended) In a procedural programming environment, a method of generating a stateful web service, the method comprising:

reading by a compiler executing on one or more computing devices a segment of procedural source code representing at least a portion of the web service;

parsing by the compiler on one or more computing devices the segment of source code to identify the presence of one or more declarative annotations identifying an associated method within the segment as being stateful;

generating on one or more computing devices one or more object codes defining one or more publicly accessible service components based at least in part upon the source code;

generating on one or more computing devices meta-data based at least in part upon the one or more declarative annotations;

associating on one or more computing devices the meta-data with the one or more object codes; and

if the presence of the one or more declarative annotations are identified by the parsing, generating, in response, on one or more computing devices one or more persistent components to maintain conversational state relating the associated method;

wherein the one or more declarative annotations further identify to the compiler the associated method as being at least one of a start method, a continue method, or a finish method, wherein the start method applies to start of a stateful conversation between a client and the web service, the continue method applies to continuation of an ongoing stateful conversation between a client and the web service, and the finish method applies to completion of an ongoing stateful conversation between a client and the web service.

17. (Previously Presented) The method of claim 16, further comprising instantiating on one or more computing devices one or more queues to temporarily store one or more requests for the identified method based at least in part upon the one or more declarative annotations further identifying the associated method as being a buffered method.

18.-19. (Cancelled)

20. (Previously Presented) The method of claim 16, further comprising:
generating on one or more computing devices a proxy object designed to facilitate interaction by the web service with an external web service.

21. (Previously Presented) The method of claim 20, further comprising:
generating on one or more computing devices a unique identifier to identify a specific instance of the external web service.

22. (Original) The method of claim 16, wherein the source code is written in the Java programming language.

23.-30. (Cancelled)

31. (Previously Presented) An article of manufacture comprising:
a storage medium; and

a plurality of programming instructions stored on the storage medium and configured to provide an integrated development environment to facilitate a user in providing input associated with web service logic of a stateful web service, and automatically specify, in response to the user input, one or more declarative annotations within the web service logic, the declarative annotations associated with an identified method of the web service logic, the declarative annotations, when recognized by a compiler through analysis of the web service logic, causing the compiler to generate one or more persistent components to maintain conversational state related to the identified method; wherein the one or more declarative annotations indicate to the compiler whether the identified method is at least one of a start method, a continue method, or a finish method, wherein the start method applies to start of a stateful conversation between a client and the web service, the continue method applies to continuation of an ongoing stateful conversation between a client and the web service, and the finish method applies to completion of an ongoing stateful conversation between a client and the web service.

32. (Previously Presented) The article of claim 31, wherein the one or more declarative annotations are specified within a source code representation of at least a portion of the web service based at least in part upon graphical input by the user.

33. (Original) The article of claim 32, wherein the one or more declarative annotations are specified within a comment field preceding the identified method.

34. (Cancelled)

35. (Original) The article of claim 31, wherein the one or more declarative annotations cause the compiler to generate a proxy object designed to facilitate interaction by the web service with one of an external web service or client.

36. (Original) The article of claim 35, wherein the one or more declarative annotations further cause the compiler to route asynchronous responses from the external web service to code specified by a developer of the web service.

37. (Original) The article of claim 35, wherein the one or more declarative annotations further cause the compiler to generate a unique identifier to identify a specific conversational instance of the external service.

38. (Currently Amended) An article of manufacture comprising:

a storage medium having stored therein a plurality of programming instructions designed to program an apparatus to generate a stateful web service, which programming instructions when executed enable the apparatus to:

read a segment of procedural source code representing at least a portion of the web service, using a compiler;

parse the segment of source code to identify the presence of one or more declarative annotations identifying an associated method within the segment as being stateful, using the compiler;

generate one or more object codes defining one or more publicly accessible service components based at least in part upon the source code;

generate meta-data based at least in part upon the one or more declarative annotations;

associate the meta-data with the one or more object codes, and

if the presence of the one or more declarative annotations is identified by said parse, generate, in response, on one or more computing devices one or more persistent components to maintain conversational state relating the associated method;

wherein the one or more declarative annotations further identify to the compiler the associated method as being at least one of a start method, a continue method, or a finish method, wherein the start method applies to start of a stateful conversation between a client and the web service, the continue method applies to continuation of an ongoing stateful conversation between a client and the web service, and the finish method applies to completion of an ongoing stateful conversation between a client and the web service.

39. (Previously Presented) The article of claim 38, wherein the instructions when executed further enable the apparatus to instantiate one or more queues to temporarily store one or more requests for the identified method based at least in part upon the one or more declarative annotations further identifying the associated method as being a buffered method.

40.-41. (Cancelled)

42. (Original) The article of claim 38, wherein the instructions when executed further enable the apparatus to generate a proxy object designed to facilitate interaction by the web service with an external web service.

43. (Original) The article of claim 42, wherein the instructions when executed further enable the apparatus to generate a unique identifier to identify a specific instance of the external web service.

44. (Original) The article of claim 38, wherein the source code is written in the Java programming language.

45.-52. (Cancelled)